

## REMARKS

In the Action dated April 17, 2007, the Examiner rejected Claims 1-7, 9-17 and 19-20 under 35 U.S.C. §103(a) as being obvious over Applicants' admitted prior art in view of Armangau. Claim 8 was rejected under §103(a) as being obvious over Applicants' admitted prior art in view of Armangau and Romer. Claim 18 was rejected under §103(a) as being obvious over Applicants' admitted prior art in view of Armangau and Evans.

In setting forth the §103(a) rejections, the Office Action states that the admitted prior art discloses the step of instructing a memory controller to move a plurality of virtual memory pages, referring to page 5, lines 16-17 of the specification, and asserting that "the 'processor' is analogous to the 'memory controller.'" Applicants respectfully disagree. A processor of a computer system is not the same thing as a memory controller, and one skilled in the art would not consider these two different components to be analogous. The prior art computer system shown in Applicants' Figure 1 helps illustrate this point. The processor is reference numeral 22 located within the processing unit, but the memory controller is part of the system memory which is reference numeral 16. Page 10, lines 4-5, of Applicants' specification similarly explains that the memory subsystem includes a memory controller and system memory. Every computer scientist understands that the processing unit is not a part of the memory subsystem, and there is absolutely no objective basis to support such an analogy. A processor handles generalized instructions such as arithmetic operations (floating-point or fixed-point) using various registers, as well as load/store operations, to access both memory systems and I/O devices. In contrast, a memory controller is limited to managing its associated memory array, and a memory controller is responsive to processor instructions. These two components have different functionalities and are clearly not interchangeable. The Office Action thus disregards the plain meaning of the explicitly recited term "memory controller."

The technique described at page 5, lines 13-17, is a software procedure as opposed to the hardware-based method of the present invention. According to that prior art technique, the operating system (OS) figures out what instructions are necessary to implement a new virtual superpage mapping, and dispatches that series of instructions to the processor which then carries

out those instructions to copy pages to new locations in physical memory corresponding to the virtual superpage. The specification even notes at page 5, line 11-12, that this “solution resorts to software-directed memory copying.” Software-based page migration has many disadvantages, among them the extra overhead necessary to direct the copying which interferes with processing performance. The OS must first utilize processor resources to derive appropriate copy instructions, and then further keeps the processor busy carrying out those instructions. More significantly (and as discussed below with regard to Armangau), the new memory addresses cannot be used until after all of the copying is completed, at which time the page table entries are coalesced into the superpage. During this wait, the application continues to suffer from poor translation-lookaside buffer (TLB) behavior (see page 5, lines 18-20).

In contrast, the present invention is achieved through hardware controls without the need for dispatching such a series of instructions to the processor, thereby reducing processing overhead. The hardware support is achieved using the memory controller which directly supervises the page migration, hence the title of the present invention (“HARDWARE SUPPORT FOR SUPERPAGE COALESCING”). The memory controller receives simplified instructions from the OS which define the set of pages to be copied, and a state engine within the memory controller uses direct memory access to carry out copying (see page 10, line 25 through page 11, line 17). The OS can use the new mappings right away because the memory controller maintains a temporary mapping from the new physical pages back to the old physical pages until the new physical pages are brought up to date. As a result, the application TLB behavior improves immediately. Since the new addresses can be used even before copying is complete, the copying can further be carried out opportunistically, imparting even more operational efficiency to the memory subsystem.

The Office Action acknowledges that the admitted prior art does not disclose accessing the virtual superpage using the new mapping while the memory controller is copying pages, and asserts that this feature is taught by Armangau. This assertion has been addressed by Applicants in their previous response which is hereby incorporated, and Applicants would respectfully reiterate that Armangau has nothing to do with the page migration or the creation of a virtual superpage, and further does not use new page mappings while copying is still underway.

Armangau is directed to a technique for backing up and restoring data in case of a storage system failure. The backup/restore process involves copying data from location A to location B then, at a later time when recovery becomes necessary, re-copying data back from location B to location A. The “snapshot copy volume” of Armangau mentioned in the Office Action is merely this backup version of data. The snapshot copy volume is not a new page mapping, that is, it does not take a set of old pages and coalesce them into a new superpage with different (more efficient) page addresses; it is just a reproduction of the data. The Office Action thus again disregards the meaning of the explicitly recited term “new virtual superpage mapping.”

Moreover, if any process is required to access data in the production volume, it does so using the original memory locations (A) for the data, not by accessing the backup location (B). The Office Action asserts that “the read/write access during snapshot maintenance is analogous to access operations while copying,” but any such read/write access will use the original memory locations (the production set), not the ostensible new mapping corresponding to the snapshot volume. The specific relevant language of Armangau at column 2, lines 16-18, states that “the production data set is accessible to a host processor for read/write access during maintenance of the snapshot copy.” Therefore, the accesses are made to “the production data set” only, and the text of Armangau never says that processor instructions use addresses of the snapshot copy. The analogy of the snapshot copy to a new virtual superpage mapping thus fails for many reasons.

Each of independent Claims 1, 7 and 14 recite memory controller-managed copying and accessing a virtual superpage using new page mappings before copying has been completed. Since the proposed combination of the admitted prior art and Armangau still fails to result in these features, it accordingly cannot render the present invention obvious.

These differences between the present invention and Armangau are also reflected in the dependent claims. Claims 3 and 9 specify that a read operation for an address of the new page mapping (which is currently being copied) is actually handled using the corresponding address of the old page mapping. In rejecting these claims the Office Action refers solely to Armangau column 2, lines 16-18. That text of Armangau is quoted above, and says only that the production set is accessible during maintenance of the snapshot copy. There is no discussion about a read

operation, or about using an old page mapping corresponding to a read address. This feature of the invention allows the read access to complete immediately without checking to see if the copying to the new virtual superpage is complete, and likewise without having to wait for the copying to complete.

Claims 6 and 16 specify updating an entry in a cache memory by modifying an address tag according to the new page mapping. In rejecting these claims, the Office Action refers primarily to Armangau column 10, lines 50-54. That portion of Armangau, however, says nothing about modifying a cache address tag according to a new page memory, but rather simply states the basic function of a cache system, *viz.*, the port adapter checking the cache memory first to see if a requested value resides there before accessing the other data storage devices. There is absolutely no discussion in Armangau of modifying cache address tags responsive to the construction of a new virtual superpage. This feature of the present invention allows the superpage construction to complete without having to flush the contents of any affected cache memory, thereby improving the overall efficiency of the memory subsystem.

Claims 12 and 13 refer to the state engine within the memory access device that reads the matching old and new page addresses, and the use of a direct memory access engine that carries out the copying, controlled by the state engine. In rejecting these claims, the Office Action states that “the ‘primary data storage subsystem’ provides the functionality of a ‘state engine,’” and that “the ‘snapshot copy facility’ is analogous to the ‘DMA engine.’” However, the primary data storage subsystem is never described as reading paired old and new page addresses from a mapping table. In this regard the Office Action refers to Armangau (i) column 6, lines 42-48, (ii) column 8, lines 4-9, and (iii) column 13, lines 17-28. The first of these references states that: “In response to a backup command from the host 20, the primary data storage subsystem 21 accesses a primary directory 26 to find data of the physical storage unit specified by the backup command in order to initiate a process of copying the data from the primary storage 27 to the second storage 29 ....” This text says nothing about sequentially reading paired old and new page addresses. The second of these references states: “The snapshot copy facility 69 includes a stored program that is executed by data processors in the primary data storage subsystem as described below with reference to Figs. 5-8. This stored program is a component of what is

known as microcode for the primary data storage subsystem.” This text does not describe direct-memory access. The snapshot copy facility is a program, i.e., software, but a DMA engine is hardware. As noted in Applicants’ specification at page 11, lines 9-11, the DMA engine is hidden from the operating system to carry out the copying and runs in the background, that is, without interrupting program processing. In contrast, the snapshot copy facility “is executed by data processors ....” The third of these references states in relevant part that the “primary data storage subsystem responds by performing a snapshot copy, and transferring backup data from the snapshot copy to the secondary storage subsystem.” This portion of Armangau again does not mention any direct-memory access functionality. The snapshot copy facility does not itself complete actual copying but rather is used to create the intermediate snapshot copy which is part of the larger backup procedure performed by the primary data storage subsystem as it transfers backup data from the snapshot copy to the secondary storage subsystem. These features of the invention allow the copying to be carried out opportunistically as noted in Applicants’ specification at page 11, lines 11-14.

Claim 15 specifies that the processing unit includes a processor core having a TLB whose entries are updated prior to completion of page copying. The Office Action states that the updating of the TLB is taught by Armangau at column 7, lines 49-64. This text discusses the de-allocation of primary storage locations to make them available for storing modified versions of physical storage units. The text does not refer to a processing unit, or a processing core, or a translation-lookaside buffer. The text further does not say anything about addresses being updated prior to completion of copying. As noted above, this feature allows the application TLB behavior to improve immediately.

For all of the foregoing reasons, Applicants respectfully request reconsideration of the §103(a) rejections.

Applicants have made a diligent effort to advance the prosecution of this application by pointing out with particularity how the claims as presented patentably define the invention over the prior art of record. In view of the remarks set forth herein, the application is believed to be in

condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the examiner is requested to telephone the undersigned.

Respectfully submitted,

/Jack V. Musgrove/

Jack V. Musgrove  
Attorney for Applicant(s)  
Reg. No. 31,986  
Telephone: 512-689-6116